

---

**tacacs<sub>plus</sub>**  
***Release alpha***

**Aug 20, 2018**



<b>1</b>	<b>TACACS+ Python client</b>	<b>1</b>
1.1	Basic Installation and Usage . . . . .	1
1.2	Programmatic Usage . . . . .	3
<b>2</b>	<b>API documentation</b>	<b>5</b>
2.1	<code>tacacs_plus.client</code> module . . . . .	5
2.2	<code>tacacs_plus.packet</code> module . . . . .	7
2.3	<code>tacacs_plus.authentication</code> module . . . . .	7
2.4	<code>tacacs_plus.authorization</code> module . . . . .	8
2.5	<code>tacacs_plus.accounting</code> module . . . . .	8
2.6	<code>tacacs_plus.flags</code> module . . . . .	9
<b>3</b>	<b>Indices and tables</b>	<b>11</b>



# CHAPTER 1

---

## TACACS+ Python client

---

A TACACS+ client that supports authentication, authorization and accounting.

Unlike RADIUS, which was designed for similar purposes, the TACACS+ protocol offers basic packet encryption but, as with most crypto designed back then, it's **not secure** and definitely should not be used over untrusted networks.

This package has been successfully used with the free **tac\_plus** TACACS+ server on a variety of operating systems.

### 1.1 Basic Installation and Usage

```
$ pip install tacacs_plus

$ tacacs_client -u myuser -H localhost authenticate
$ tacacs_client -u myuser -H localhost authenticate -t pap
$ tacacs_client -u myuser -H localhost -v authenticate -t chap
status: PASS

$ tacacs_client -u myuser -H localhost authorize -c service=shell cmd=show_
↪cmdarg=version
$ tacacs_client -u myuser -H localhost -v authorize -t pap -c service=shell cmd=show_
↪cmdarg=version
status: PASS

$ tacacs_client -u myuser -H localhost -v authorize -t pap -c service=junos-exec
status: REPL
av-pairs:
    allow-commands=^acommandregex$
    deny-commands=^anothercommandregex$

$ tacacs_client -u myuser -H localhost account -f start -c service=shell cmd=show_
↪cmdarg=version
$ tacacs_client -u myuser -H localhost account -f stop -c service=shell cmd=show_
↪cmdarg=version
```

(continues on next page)

(continued from previous page)

```

$ tacacs_client -h
usage: tacacs_client [-h] -u USERNAME -H HOST [-p PORT] [-l PRIV_LVL]
                    [-t {ascii,pap,chap}] [-r REM_ADDR] [-P VIRTUAL_PORT]
                    [--timeout TIMEOUT] [-d] [-v] [-k KEY]
                    {authenticate,authorize,account} ...

    Tacacs+ client with full AAA support:

    * Authentication supports both ascii, pap and chap.
    * Authorization supports AV pairs and single commands.
    * Accounting support AV pairs and single commands.

    NOTE: shared encryption key can be set via environment variable TACACS_PLUS_
    ↪KEY or via argument.
    NOTE: user password can be setup via environment variable TACACS_PLUS_PWD or
    ↪via argument.

positional arguments:
  {authenticate,authorize,account}
                                action to perform over the tacacs+ server
  authenticate             authenticate against a tacacs+ server
  authorize                 authorize a command against a tacacs+ server
  account                  account commands with accounting flags against a tacacs+
  ↪server

optional arguments:
  -h, --help                show this help message and exit
  -u USERNAME, --username USERNAME
                                user name
  -H HOST, --host HOST      tacacs+ server address
  -p PORT, --port PORT      tacacs+ server port (default 49)
  -l PRIV_LVL, --priv-lvl PRIV_LVL
                                user privilege level
  -t {ascii,pap,chap}, --authen-type {ascii,pap,chap}
                                authentication type
  -r REM_ADDR, --rem-addr REM_ADDR
                                remote address (logged by tacacs server)
  -P VIRTUAL_PORT, --virtual-port VIRTUAL_PORT
                                console port used in connection (logged by tacacs server)
  --timeout TIMEOUT
  -d, --debug                enable debugging output
  -v, --verbose              print responses
  -k KEY, --key KEY          tacacs+ shared encryption key

$ tacacs_client authenticate -h
usage: tacacs_client authenticate [-h] [-p PASSWORD]

optional arguments:
  -h, --help                show this help message and exit
  -p PASSWORD, --password PASSWORD
                                user password

$ tacacs_client authorize -h
usage: tacacs_client authorize [-h] -c CMDS [CMDS ...]

```

(continues on next page)

(continued from previous page)

```

optional arguments:
  -h, --help            show this help message and exit
  -c CMDS [CMDS ...], --cmds CMDS [CMDS ...]
                        list of cmds to authorize

$ tacacs_client account -h
usage: tacacs_client account [-h] -c CMDS [CMDS ...] -f {start,stop,update}

optional arguments:
  -h, --help            show this help message and exit
  -c CMDS [CMDS ...], --cmds CMDS [CMDS ...]
                        list of cmds to authorize
  -f {start,stop,update}, --flag {start,stop,update}
                        accounting flag

```

## 1.2 Programmatic Usage

```

#!/usr/bin/env python
from tacacs_plus.client import TACACSClient
from tacacs_plus.flags import TAC_PLUS_ACCT_FLAG_START, TAC_PLUS_ACCT_FLAG_WATCHDOG,
↪TAC_PLUS_ACCT_FLAG_STOP

cli = TACACSClient('host', 49, 'secret', timeout=10)

# authenticate user and pass
authen = cli.authenticate('username', 'password')
print "PASS!" if authen.valid else "FAIL!"

# authorize user and command
author = cli.authorize('username', arguments=[b"service=shell", b"cmd=show", b
↪"cmdargs=version"])
print "PASS!" if author.valid else "FAIL!"

# start accounting session for command
acct = cli.account('username', TAC_PLUS_ACCT_FLAG_START, arguments=[b"service=shell",
↪b"cmd=show", b"cmdargs=version"])
print "PASS!" if acct.valid else "FAIL!"

# continue accounting session for another command
acct = cli.account('username', TAC_PLUS_ACCT_FLAG_WATCHDOG, arguments=[b"service=shell
↪", b"cmd=debug", b"cmdargs=aaa"])
print "PASS!" if acct.valid else "FAIL!"

# close accounting session
acct = cli.account('username', TAC_PLUS_ACCT_FLAG_STOP, arguments=[b"service=shell", b
↪"cmd=exit"])
print "PASS!" if acct.valid else "FAIL!"

```





## CHAPTER 2

---

### API documentation

---

This is the tacacs\_plus API documentation. It contains the documentation extracted from the docstrings of the various classes, methods, and functions in the tacacs\_plus package. If you want to know what a certain function/method does, this is the place to look.

#### Contents

- *API documentation*
  - *tacacs\_plus.client module*
  - *tacacs\_plus.packet module*
  - *tacacs\_plus.authentication module*
  - *tacacs\_plus.authorization module*
  - *tacacs\_plus.accounting module*
  - *tacacs\_plus.flags module*

## 2.1 tacacs\_plus.client module

source: [https://github.com/ansible/tacacs\\_plus/blob/master/tacacs\\_plus/client.py](https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/client.py)

**class** tacacs\_plus.client.**TACACSClient** (*host, port, secret, timeout=10, session\_id=None, family=2, version\_max=12, version\_min=0*)

A TACACS+ authentication client. <https://datatracker.ietf.org/doc/draft-ietf-opsawg-tacacs>

An open source TACACS+ server daemon is available at [http://www.shrubbery.net/tac\\_plus/](http://www.shrubbery.net/tac_plus/)

**account** (*username, flags, arguments=[], authen\_type=1, priv\_lvl=0, rem\_addr='python\_device', port='python\_tty0'*)

Account with a TACACS+ server.

**Parameters**

- **username** –
- **flags** – TAC\_PLUS\_ACCT\_FLAG\_START, TAC\_PLUS\_ACCT\_FLAG\_WATCHDOG, TAC\_PLUS\_ACCT\_FLAG\_STOP
- **arguments** – The authorization arguments
- **authen\_type** – TAC\_PLUS\_AUTHEN\_TYPE\_ASCII, TAC\_PLUS\_AUTHEN\_TYPE\_PAP, TAC\_PLUS\_AUTHEN\_TYPE\_CHAP
- **priv\_lvl** – Minimal Required priv\_lvl.
- **rem\_addr** – AAA request source, default to TAC\_PLUS\_VIRTUAL\_REM\_ADDR
- **port** – AAA port, default to TAC\_PLUS\_VIRTUAL\_PORT

**Returns** TACACSAccountingReply**Raises** socket.timeout, socket.error

**authenticate** (*username*, *password*, *priv\_lvl*=0, *authen\_type*=1, *chap\_ppp\_id*=None, *chap\_challenge*=None, *rem\_addr*='python\_device', *port*='python\_tty0')

Authenticate to a TACACS+ server with a username and password.

**Parameters**

- **username** –
- **password** –
- **priv\_lvl** –
- **authen\_type** – TAC\_PLUS\_AUTHEN\_TYPE\_ASCII, TAC\_PLUS\_AUTHEN\_TYPE\_PAP, TAC\_PLUS\_AUTHEN\_TYPE\_CHAP
- **chap\_ppp\_id** – PPP ID when authen\_type == 'chap'
- **chap\_challenge** – challenge value when authen\_type == 'chap'
- **rem\_addr** – AAA request source, default to TAC\_PLUS\_VIRTUAL\_REM\_ADDR
- **port** – AAA port, default to TAC\_PLUS\_VIRTUAL\_PORT

**Returns** TACACSAuthenticationReply**Raises** socket.timeout, socket.error

**authorize** (*username*, *arguments*=[], *authen\_type*=1, *priv\_lvl*=0, *rem\_addr*='python\_device', *port*='python\_tty0')

Authorize with a TACACS+ server.

**Parameters**

- **username** –
- **arguments** – The authorization arguments
- **authen\_type** – TAC\_PLUS\_AUTHEN\_TYPE\_ASCII, TAC\_PLUS\_AUTHEN\_TYPE\_PAP, TAC\_PLUS\_AUTHEN\_TYPE\_CHAP
- **priv\_lvl** – Minimal Required priv\_lvl.
- **rem\_addr** – AAA request source, default to TAC\_PLUS\_VIRTUAL\_REM\_ADDR
- **port** – AAA port, default to TAC\_PLUS\_VIRTUAL\_PORT

**Returns** TACACSAuthenticationReply

**Raises** socket.timeout, socket.error

**closing** (*\*\*kws*)

**send** (*body, req\_type, seq\_no=1*)

Send a TACACS+ message body

**Parameters**

- **body** – packed bytes, i.e., *struct.pack(...)*
- **req\_type** – TAC\_PLUS\_AUTHEN, TAC\_PLUS\_AUTHOR, TAC\_PLUS\_ACCT
- **seq\_no** – The sequence number of the current packet. The first packet in a session **MUST** have the sequence number 1 and each subsequent packet will increment the sequence number by one. Thus clients only send packets containing odd sequence numbers, and TACACS+ servers only send packets containing even sequence numbers.

**Returns** TACACSPacket

**Raises** socket.timeout, socket.error

**sock**

**version**

## 2.2 tacacs\_plus.packet module

source: [https://github.com/ansible/tacacs\\_plus/blob/master/tacacs\\_plus/packet.py](https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/packet.py)

**class** tacacs\_plus.client.TACACSHeader (*version, type, session\_id, length, seq\_no=1, flags=0*)

**packed**

**classmethod** unpacked (*raw*)

**version\_max**

**version\_min**

**class** tacacs\_plus.client.TACACSPacket (*header, body\_bytes, secret*)

**body**

**crypt**

**encrypted**

**seq\_no**

## 2.3 tacacs\_plus.authentication module

source: [https://github.com/ansible/tacacs\\_plus/blob/master/tacacs\\_plus/authentication.py](https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/authentication.py)

**class** tacacs\_plus.authentication.TACACSAuthenticationStart (*username, auth\_type, priv\_lvl=0, data="", rem\_addr='python\_device', port='python\_tty0'*)

```
    packed

class tacacs_plus.authentication.TACACSAuthenticationContinue (password,
                                                                data="", flags=0)

    packed

class tacacs_plus.authentication.TACACSAuthenticationReply (status,          flags,
                                                            server_msg, data)

    error
    getpass
    human_status
    invalid
    classmethod unpacked (raw)
    valid
```

## 2.4 tacacs\_plus.authorization module

source: [https://github.com/ansible/tacacs\\_plus/blob/master/tacacs\\_plus/authorization.py](https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/authorization.py)

```
class tacacs_plus.authorization.TACACSAuthorizationStart (username,          au-
                                                            then_method,  priv_lvl,
                                                            authn_type,  arguments,
                                                            rem_addr='python_device',
                                                            port='python_tty0')

    packed

class tacacs_plus.authorization.TACACSAuthorizationReply (status,          arg_cnt,
                                                            server_msg,    data,
                                                            arguments)

    error
    follow
    human_status
    invalid
    reply
    classmethod unpacked (raw)
    valid
```

## 2.5 tacacs\_plus.accounting module

source: [https://github.com/ansible/tacacs\\_plus/blob/master/tacacs\\_plus/accounting.py](https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/accounting.py)

```
class tacacs_plus.accounting.TACACSAccountingStart (username, flags, authen_method,
                                                    priv_lvl, authen_type, arguments,
                                                    rem_addr='python_device',
                                                    port='python_tty0')

    packed

class tacacs_plus.accounting.TACACSAccountingReply (status, server_msg, data)

    error
    follow
    human_status
    classmethod unpacked (raw)
    valid
```

## 2.6 tacacs\_plus.flags module

source: [https://github.com/ansible/tacacs\\_plus/blob/master/tacacs\\_plus/flags.py](https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/flags.py)

this module contains all the constant flags used to implement the tacacs+ RFC.



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`





## A

account() (tacacs\_plus.client.TACACSCClient method), 5  
 authenticate() (tacacs\_plus.client.TACACSCClient method), 6  
 authorize() (tacacs\_plus.client.TACACSCClient method), 6

## B

body (tacacs\_plus.client.TACACSPacket attribute), 7

## C

closing() (tacacs\_plus.client.TACACSCClient method), 7  
 crypt (tacacs\_plus.client.TACACSPacket attribute), 7

## E

encrypted (tacacs\_plus.client.TACACSPacket attribute), 7  
 error (tacacs\_plus.accounting.TACACSAccountingReply attribute), 9  
 error (tacacs\_plus.authentication.TACACSAuthenticationReply attribute), 8  
 error (tacacs\_plus.authorization.TACACSAuthorizationReply attribute), 8

## F

follow (tacacs\_plus.accounting.TACACSAccountingReply attribute), 9  
 follow (tacacs\_plus.authorization.TACACSAuthorizationReply attribute), 8

## G

getpass (tacacs\_plus.authentication.TACACSAuthenticationReply attribute), 8

## H

human\_status (tacacs\_plus.accounting.TACACSAccountingReply attribute), 9  
 human\_status (tacacs\_plus.authentication.TACACSAuthenticationReply attribute), 8

human\_status (tacacs\_plus.authorization.TACACSAuthorizationReply attribute), 8

## I

invalid (tacacs\_plus.authentication.TACACSAuthenticationReply attribute), 8  
 invalid (tacacs\_plus.authorization.TACACSAuthorizationReply attribute), 8

## P

packed (tacacs\_plus.accounting.TACACSAccountingStart attribute), 9  
 packed (tacacs\_plus.authentication.TACACSAuthenticationContinue attribute), 8  
 packed (tacacs\_plus.authentication.TACACSAuthenticationStart attribute), 7  
 packed (tacacs\_plus.authorization.TACACSAuthorizationStart attribute), 8  
 packed (tacacs\_plus.client.TACACSHeader attribute), 7

## R

reply (tacacs\_plus.authorization.TACACSAuthorizationReply attribute), 8

## S

send() (tacacs\_plus.client.TACACSCClient method), 7  
 seq\_no (tacacs\_plus.client.TACACSPacket attribute), 7  
 sock (tacacs\_plus.client.TACACSCClient attribute), 7

## T

TACACSAccountingReply	(class	in
tacacs_plus.accounting), 9		
TACACSAccountingStart	(class	in
tacacs_plus.accounting), 8		
TACACSAuthenticationContinue	(class	in
tacacs_plus.authentication), 8		
TACACSAuthenticationReply	(class	in
tacacs_plus.authentication), 8		

TACACSAuthenticationStart (class in  
tacacs\_plus.authentication), 7  
TACACSAuthorizationReply (class in  
tacacs\_plus.authorization), 8  
TACACSAuthorizationStart (class in  
tacacs\_plus.authorization), 8  
TACACSClient (class in tacacs\_plus.client), 5  
TACACSHeader (class in tacacs\_plus.client), 7  
TACACSPacket (class in tacacs\_plus.client), 7

## U

unpacked() (tacacs\_plus.accounting.TACACSAccountingReply  
class method), 9  
unpacked() (tacacs\_plus.authentication.TACACSAuthenticationReply  
class method), 8  
unpacked() (tacacs\_plus.authorization.TACACSAuthorizationReply  
class method), 8  
unpacked() (tacacs\_plus.client.TACACSHeader class  
method), 7

## V

valid (tacacs\_plus.accounting.TACACSAccountingReply  
attribute), 9  
valid (tacacs\_plus.authentication.TACACSAuthenticationReply  
attribute), 8  
valid (tacacs\_plus.authorization.TACACSAuthorizationReply  
attribute), 8  
version (tacacs\_plus.client.TACACSClient attribute), 7  
version\_max (tacacs\_plus.client.TACACSHeader at-  
tribute), 7  
version\_min (tacacs\_plus.client.TACACSHeader at-  
tribute), 7